

The Design Recipe

Lesson time: 30-60 Minutes

LESSON OVERVIEW

In the last stage, students wrote some very simple functions - but more sophisticated functions demand a more thoughtful approach. The Design Recipe is a structured approach to writing functions that includes writing test cases to ensure that the function works as expected. Once students have mastered the Design Recipe process, they can apply it to any word problem they encounter.

LESSON OBJECTIVES

Students will:

- Use the Design Recipe to identify dependent variables, independent variables, and constants.

ANCHOR STANDARD

Common Core Math Standards

- **F.BF.1:** Write a function that describes a relationship between two quantities.

Additional standards alignment can be found at the end of this lesson

TEACHING SUMMARY

Getting Started

- 1) Vocabulary
- 2) What is the Design Recipe

Activity: The Design Recipe

- 3) Collaborative Design

TEACHING GUIDE

MATERIALS, RESOURCES, AND PREP

For the Student

- Fast Functions Sheet
- Blank Design Recipe Form

For the Teacher

- Lesson Slide Deck

GETTING STARTED

1) Vocabulary

This lesson has five new and important words:

- **Design Recipe** - a sequence of steps to document, test, and write functions.
- **Purpose Statement** - a brief description of what the function does.
- **Independent Variable** - The value that the experimenter controls. The input.
- **Dependent Variable** - The value that changes based on the independent variable. The output.
- **Constant** - A fixed number in a relationship.

2) What is the Design Recipe

The Design Recipe is a roadmap for defining functions, which programmers use to make sure the code they write does what they want it to do. Each step builds on the last, so any mistakes can be caught early in the process. This roadmap has a series of steps:

1. Write a Contract that describes the word problem.
2. Write Examples based on the contract.
3. Define a function that matches the examples.

Let's start out by applying the Design Recipe together to the following problem:

Define a function 'purple-star', that takes in the size of the star and produces an outlined, purple star of the given size.

Step 1 - The Contract





purple-star: Number -> Image

Be sure to include a good Name for each function, and remember that the Domain and Range can only include types like Numbers, Images, Strings, etc.





A Contract is the foundation for a function, which gives programmers just enough information to use it: the name of the function, the type (or types) of data it expects and the type of data it returns.

Step 2 - Examples

Example (call, result)

purple_star (size)	star (radius, style, color)		
			

Example (call, result)

purple_star (size)	star (radius, style, color)		
			

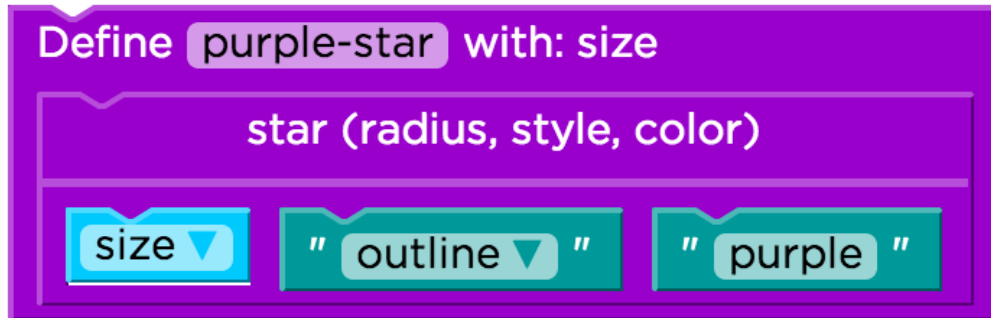
- Every Example begins with the name of the function. Where could you find the name of the function?
- Every Example has to include sample inputs. Where could you find out how many inputs this function needs,

and what types they are?

- Every Example has to include an expression for what the function should do when given an input. Where could you look to find out what this function does?

Once you have two or more Examples, it should be easy to identify what has changed between them. In fact, the number of things that change should match the number of things in the function's Domain: if the Domain has a Number and a String in it, then those two values should be the things that differ between your Examples.

Step 3 - Function Definition



By identifying what has changed between these Examples, we can define our actual function.

Challenge students to explain why this function does not need to know the color of the star, or whether or not it is solid. The main idea here is that the function already "knows" these things, so the only thing that is changing is the size of the star.

Remember that the Contract and Purpose Statement can be used to write the Examples, even if a programmer isn't sure how to begin.

ACTIVITIES:

3) Collaborative Design

- Define a function 'spot', that takes in a color and produces a solid circle of radius 50, filled in with that color.
- Define a function 'average', which takes in two numbers and produces their average. (You may need to remind the students that to find the average of two numbers, they should be added together and divided by two.)
- Suppose a company logo is a word drawn in big, red letters, rotated some number of degrees. Define a function 'logo', that takes in a company name and a rotation, and produces a logo for that company.

Put students into groups of 3 - each member of the group will represent one step of the Design Recipe

1. Contract
2. Examples
3. Function

Each group will work through a set of word problems using the [Fast Functions Sheet](#). We recommend that you pull word problems from your own curriculum so that students can see how the Design Recipe can be used outside of programming. Make sure that each group member stays true to their role and that they work through the steps in the right order. If you don't have problems to use from your curriculum, there are a number of examples available in this lesson's slide deck.

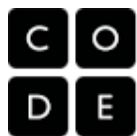
LESSON TIP

Challenge students to explain their Examples (their function name, the number of inputs, their types and the type of the returned value). Make sure that the two Examples for each function have different input values! For each of these questions, students must be able to point to the specific part of their Contract as the justification for their Example.

Make sure students have chosen good variable names for their function definitions, and ask students to justify every part of the function body. The only acceptable answers should be "I copied this because it's the same in both Examples", or "I used

a variable name because it differs between Examples."

Once students have worked through the Fast Functions, you can have them use the full Blank Design Recipe Form to work through an word problems that they encounter in the future.



Code.org UK



Derived from
BOOTSTRAP
www.bootstrapworld.org

This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0)

If you are interested in licensing Code.org materials for commercial purposes, contact us: <https://code.org/contact>